

Constructing SAT Filters with a Quantum Annealer

Adam Douglass, Andrew D. King^(✉), and Jack Raymond

D-Wave Systems Inc., 3033 Beta Avenue,
Burnaby, BC V5G 4M9, Canada
{adouglass,aking,jraymond}@dwavesys.com

Abstract. SAT filters are a novel and compact data structure that can be used to quickly query a word for membership in a fixed set. They have the potential to store more information in a fixed storage limit than a Bloom filter. Constructing a SAT filter requires sampling diverse solutions to randomly constructed constraint satisfaction instances, but there is flexibility in the choice of constraint satisfaction problem. Presented here is a case study of SAT filter construction with a focus on constraint satisfaction problems based on MAX-CUT clauses (Not-all-equal 3-SAT, 2-in-4-SAT, etc.) and frustrated cycles in the Ising model. Solutions are sampled using a D-Wave quantum annealer, and results are measured against classical approaches. The SAT variants studied are of interest in the context of SAT filters, independent of the solvers used.

Keywords: SAT filter · Quantum annealing · Ising model · Maximum cut · Sampling · Constraint satisfaction problem

1 Introduction

Weaver et al. [37] recently presented *SAT filters* as an efficient data structure by which a set can be filtered with no false negatives and a false positive rate approaching zero as the situation requires. The construction of a SAT filter with high space efficiency and low false positive rate requires finding many solutions to a random n -variable instance of a constraint satisfaction problem (CSP), which was originally chosen to be k -SAT for $k \geq 3$ [37]. The effectiveness of the filter with respect to storage requirements, or *efficiency*, requires a low probability that a randomly generated clause would be satisfied by every solution in the filter. Consequently the solutions must satisfy various independence tests, e.g. they should not differ by only a few bitflips.

This application is a natural fit for a D-Wave Two (DW2) quantum annealer, which is capable of quickly sampling many low-energy states of a spin system in the Ising model [18, 35] using open-system quantum annealing [1, 3, 9]. As the basis of the filter one can choose a CSP that is readily expressed in the Ising model, for example Not-all-equal 3-SAT [17, 20]. In the context of filters and elsewhere, CSP solution sampling is a problem of both theoretical and practical interest [6, 16].

This paper reports on a case study using a D-Wave quantum annealer to solve SAT instances and construct SAT filters, providing comparisons with classical solvers and comparing filters generated using several SAT variants, the choice of which has great effect on the characteristics of the resulting filter. Although processor size restricts the study to relatively small instances, the study gives a first look at several “exotic” SAT filters and an early application of sampling optima via quantum annealing.

Section 2 discusses the preliminaries of SAT filters and *blocked* SAT filters, which involve constructing a SAT filter for each bucket of a dense hash table. Section 3 discusses the expression and solution of CSPs in the Ising model. Section 4 discusses the CSPs used to construct filters. Sections 5 and 6 review the results of two cases in this study; the first investigates the construction of a filter with many blocks of fixed size, while the second investigates how filters and solver performance evolve for various CSPs as the number of variables increases. Finally Section 7 offers conclusions.

2 Filters, SAT Filters and Blocked SAT Filters

Given a domain W and a subset $X \subset W$, the *set membership problem* is to determine if an element x of W is in X . A *filter* is a data structure that allows fast set membership queries, possibly with false positives but not false negatives. That is, if $x \in X$ then the filter will always return $F(x) = 1$ (*maybe*). If $x \notin X$ then the filter will (deterministically) return either $F(x) = 1$ (*maybe*) or $F(x) = 0$ (*no*). Here it is assumed that $|X| \ll |W|$, and therefore the *false positive rate* p of the filter can be defined as the probability that a randomly selected $x \in W$ has $F(x) = 1$.

The aim is to minimize the false positive rate of F while minimizing storage requirements (in bits, denoted by $|F|$), construction time, and query time. Given a storage limit of $|F|$ bits, the information-theoretic limit on the false positive rate is given by the equation

$$\frac{-\log_2(p)}{|F|/|X|} \leq 1. \quad (1)$$

This result appears as Theorem 4.1 in [37], summarizing results in [36]. Here the numerator represents the *bits of cut-down* and the denominator represents the average storage bits per keyword in X . The *efficiency* of the filter F is therefore defined as

$$\mathcal{E}(F) = \frac{-\log_2(p)}{|F|/|X|}. \quad (2)$$

The standard tool for this situation is a *Bloom filter* [2]. A Bloom filter F is constructed for X with false positive rate p using parameters r and n derived from $|X|$ and p . The filter consists of a bitstring $B = (b_1, \dots, b_n)$ and r hash functions h_1, \dots, h_r , each of which hashes an element of W (ideally uniformly) to an element of $\{1, \dots, n\}$. To construct the filter, initialize all bits of B to zero.

Then for each $x \in X$ and hash function h_i set $b_{h_i(x)} = 1$. To query $F(x)$, simply check if $\prod_{i=1}^r b_{h_i(x)} = 1$. If so, then $F(x) = 1$. If not, then $F(x) = 0$. Optimal choices of n and r will, for sufficiently large X , give a filter F with efficiency $\mathcal{E}(F) \approx \ln(2) \approx 0.69$.

2.1 SAT Filters

It is possible to exceed the efficiency of a Bloom filter using *SAT filters*, introduced recently by Weaver et al. [37]. A SAT filter is, like a Bloom filter, a space-efficient data structure used for set membership testing with one-way error¹. Unlike Bloom filters, SAT filters are *offline*: new elements cannot be added to the filter after the filter is built. And unlike any online filters [24], SAT filters can achieve efficiency arbitrarily close to 1. SAT filters can be constructed using a variety of constraints, and the constraint used typically imposes an upper bound on efficiency (see Section 4.4). This section describes a 3-SAT filter for illustrative purposes.

During construction of a Bloom filter with $r = 1$, each keyword $x \in X$ is hashed to a bit in B . During construction of a 3-SAT filter with $r = 1$, each keyword $x_i \in X$ is hashed to a uniformly chosen 3-SAT clause $C_i = h(x_i)$ over n variables, where n is given an appropriate value with respect to $|X|$. Here a clause is constructed by choosing three distinct variables at random, and negating each one independently with probability $\frac{1}{2}$. The conjunction of these random clauses gives a random 3-SAT formula:

$$\mathcal{F} = \bigwedge_i^{|X|} C_i. \quad (3)$$

Random 3-SAT instances have a satisfiability phase transition near 4.26 clauses per variable [8]. This means that if $|X| \leq (4.26 - \epsilon)n$ for $\epsilon > 0$, and $|X|$ is sufficiently large, then \mathcal{F} is satisfiable with high probability [27]. Storage of the 3-SAT filter F consists of storing the hash function h and a collection $S = (s_1, \dots, s_k)$ of solutions to \mathcal{F} . $F(x)$ is queried by checking whether the clause $C = h(x)$ is satisfied by each truth assignment in S .

If $r = 1$, then $|F| = n$ (here hash functions are not included in the analysis of storage cost, so we store $r = 1$ bits per variable) and the false positive rate of a 3-SAT filter is $7/8$. Since $|F|/|X| \gtrsim 1/4.26$, this gives the efficiency

$$\mathcal{E}(F) = \frac{-\log_2(7/8)}{n/|X|} \gtrsim 0.81. \quad (4)$$

The false positive rate given by a single solution is generally too high. Weaver et al. provided two ways of dealing with this issue [37]. The first is to construct a *multi-instance* filter, which stores one solution to each of multiple formulae. The second is to construct a *single-instance* filter, which stores multiple solutions to

¹ A variety of alternatives to Bloom filters have been proposed, e.g. [11, 28–30].

a single formula. The former suffers from increased query time, since multiple hashes must be evaluated. The latter suffers from decreased efficiency, since multiple solutions will generally not be perfectly independent. The severity of this dependence depends heavily on the SAT variant used.

2.2 Blocked SAT Filters

In certain situations it is impractical to find solutions to a SAT formula on n variables and $|X|$ clauses – this case study is limited to CSPs on at most 110 variables (typically fewer) due to the limited size of the D-Wave processor. This situation calls for a *blocked* filter [30], which uses a *blocking hash function* $\hat{h} : W \rightarrow \{1, \dots, b\}$ that divides X into b blocks

$$X_i = \{x \in X \mid \hat{h}(x) = i\}. \quad (5)$$

A filter F_i is then constructed for each block X_i , and to query $x \in W$ for membership in X , x is queried for membership in $X_{\hat{h}(x)}$ using $F_{\hat{h}(x)}$.

This blocked filter, like a normal filter, has one-way error. Under the assumption that \hat{h} maps members of W to $\{1, \dots, b\}$ equiprobably, the false positive rate p is the mean of the b false positive rates $\{p_i\}_{i=1}^b$ of the individual filters. More rigorously, letting $W_i = \{x \in W \mid \hat{h}(x) = i\}$,

$$p = \sum_{i=1}^m \frac{|W_i|}{|W|} p_i. \quad (6)$$

Even in the case where $r = 1$, blocked filters can adversely affect efficiency because of the variance in bin sizes under \hat{h} [21, 30]. This issue is considered in Section 5.

3 Solving CSPs in the Ising Model with a D-Wave Processor

This paper contains results from a D-Wave Two (DW2) quantum annealing processor using a *Washington W3* chip operating over 1097 of 1152 configured qubits in a \mathcal{C}_{12} *Chimera* layout [4]. All runs use the minimum anneal time of 20 μ s.

D-Wave quantum annealing processors are designed to minimize the energy of an Ising spin configuration. Input to the processor consists of an *Ising Hamiltonian* (h, J) , where $h \in \mathbb{R}^N$ is a vector of *local fields* and $J \in \mathbb{R}^{N \times N}$ is a matrix of *couplings*, which may be assumed to be symmetric. The *energy* of a spin configuration $s \in \{-1, 1\}^N$ is defined as

$$E(s) = E(h, J, s) = s^T J s + s^T h. \quad (7)$$

The output of an *anneal* (i.e. a hardware run) of the processor is a low-energy state s , which consists of an Ising spin (either -1 or 1) for each *qubit*.

In a D-Wave processor, not all pairs of qubits are coupled, and therefore the set of nonzero entries of J must respect the physical constraints of the processor. One can view (h, J) as a set of vertex and edge weights, respectively, of the *qubit connectivity graph*, whose vertices correspond to qubits and whose edges correspond to couplers. The qubit connectivity graph for the processor used in this report is shown in Fig. 1.

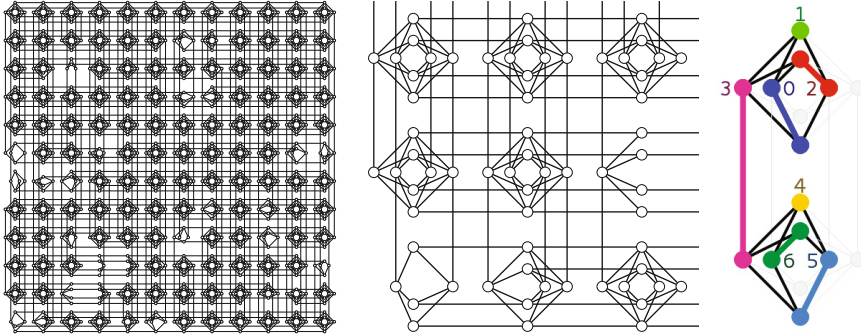


Fig. 1. The 1097-qubit hardware graph (left) of the processor used, with detail (center) of the bottom-left corner. This graph is a subgraph of the 1152-qubit 12×12 Chimera grid. The irregular pattern of connectivity was designed so that high-connectivity problems can be effectively emulated. (right, color online) An embedded problem of two interacting clauses is shown ($2in4(x_0, x_1, x_2, x_3) \wedge 2in4(x_3, x_4, x_5, x_6)$). Each variable is represented by one or more qubits; thicker lines indicate two variables acting as a single logical variable (achieved by energy penalties causing qubit behavior to coincide). Given this contraction the variables interact in two cliques through the remaining edges.

A state s minimizing $E(h, J, s)$ is called a *ground state* of (h, J) ; all other states are called *excited states*. A lowest-energy excited state is called a *first excited state*. Given a ground state s_0 and a first excited state s_1 , $g := E(s_1) - E(s_0)$ is the *minimum final gap* of the Hamiltonian.

3.1 Constraint Satisfaction Problems in the Ising Model

Let f be a Boolean function with an n -dimensional binary range. For simplicity, assume $f : \{-1, 1\}^n \rightarrow \{0, 1\}$. Now suppose an n -dimensional Ising Hamiltonian (h, J) satisfies $E(s) = x$ if $f(s) = 0$, and $E(s) \geq x + g$ if $f(s) = 1$. Then (h, J) encodes f with gap g .

Now take a collection of Boolean functions

$$\{ f_i : \{-1, 1\}^n \rightarrow \{0, 1\} \}_{i=1}^k$$

and a collection of Hamiltonians $\{(h_i, J_i)\}_{i=1}^k$ such that (h_i, J_i) encodes f_i with gap g_i . Then the Hamiltonian $(\sum_{i=1}^k h_i, \sum_{i=1}^k J_i)$ encodes $\max_i f_i$ with gap $\min_i g_i$.

This fact makes the Ising model suitable for application to constraint satisfaction problems [15, 19, 25]. In particular, it is straightforward to formulate certain SAT variants as Ising problems in such a way that they can be solved by DW2: Let $f(s) = 0$ represent satisfaction; max therefore represents conjunction in this context.

3.2 Graph Minors in the Ising Model

Random instances of the SAT variants considered here are constructed as a weighted subgraph G of either a complete graph K_n or a complete bipartite graph $K_{\frac{n}{2}, \frac{n}{2}}$. In general, these instances cannot be solved directly using the D-Wave processor's native connectivity. Rather, G must be embedded in the hardware graph G_H as a *graph minor*, which amounts to transforming G_H into G by the operations of *edge contraction*, *vertex deletion*, and *edge deletion* [10]. Edge and vertex deletion can be realized trivially in the Ising model by setting couplers to $J = 0$. Edge contraction can be realized by setting couplers to $J = -\kappa$, where κ is large, and the coupling therefore compels a set of qubits to act as a single *logical qubit*. The choice of the parameter κ is nontrivial. In this paper, runs on minor-embedded instances are optimized over five possible choices of κ . See [7, 20, 35] for further discussion. Embedding G into G_H is achieved here using a specialized heuristic algorithm [5]. In Fig. 1, edge contraction of a 2-cell Chimera graph allows representation of a 2in4SAT Ising problem described a pair of 4-cliques, anti-ferromagnetically coupled, sharing one variable.

4 SAT Variants in the Ising Model

Although k -SAT was previously used to construct SAT filters [37], it is not amenable to representation in the Ising model. To represent a SAT relations k binary variables with only pairwise relations available is not possible for $k \geq 3$ without the introduction of additional *ancillary* variables to mediate the interactions. To represent an instance on n variables and m clauses requires at least m ancillary variables, so an instance near the phase transition has many more ancillary Ising variables than original variables. Not-all-equal 3-SAT (*NAE3SAT*) is an NP-complete SAT variant that does not present the same obstacle [14, 17, 20]. This section first discusses issues surrounding NAE3SAT, then moves on to further extensions. Ultimately we are interested in sampling satisfying assignments, which for the problem classes presented is NP-hard [16].

4.1 NAE3SAT

An NAE3SAT clause consists of three literals, and is satisfied precisely if at least one literal is true and one literal is false. Thus a clause ensures that three

literals are not all equal. This requirement is easily expressed in the Ising model. Consider the Ising problem

$$\min_{s_1, s_2, s_3 \in \{-1, 1\}} (s_1 s_2 + s_2 s_3 + s_1 s_3).$$

This Hamiltonian has six ground states with energy -1 , corresponding to states in which not all variables are equal. The other two states have energy 3 , meaning that this Hamiltonian encodes a NAE3SAT clause with gap 4 . The qubit connectivity graph of a NAE3SAT clause is a triangle with a coupling of $+1$ on each edge.

Monotone NAE3SAT. Unlike 3-SAT, NAE3SAT remains NP-complete when all literals are non-negated – in this case the problem is equivalent to hypergraph 2-coloring [23]. This variant is called *monotone NAE3SAT* or *MNAE3SAT*. Asymptotically, NAE3SAT and MNAE3SAT have very similar characteristics. For small systems, however, marginal gains in efficiency can be realized from the fact that not all solutions satisfy the same number of possible clauses. These small-system considerations are relevant due to limitations imposed by the D-Wave processor. All MNAE3SAT instances studied here have 45 variables. Instances on up to around 65 variables can be embedded consistently.

4.2 MAX-CUT SAT Variants

The Ising formulation of an NAE3SAT clause can be thought of as an unweighted MAX-CUT problem on a triangle, i.e. a K_3 . Replacing K_3 with larger cliques gives a sequence of SAT variants with different properties: NAE3SAT, 2in4SAT, 2or3in5SAT, 3in6SAT, etc. These variants can more generally be called k MCSAT for $k = 3, 4, 5, 6, \dots$. A randomly constructed k MCSAT clause contains a random k -set of variables, each one of which is negated independently with probability $\frac{1}{2}$; the coupling on the edge between two variables is -1 if precisely one is negated, and is $+1$ otherwise.

When k is even, k MCSAT clauses are locally inflexible, since no single bit flip connects any two solutions. When these clauses are agglomerated in a random manner, a *global* rearrangement of the state space is required on the core of the graph to move between solutions. In other words, solutions are isolated. If the core of the graph is the graph itself (i.e. if every variable is in at least two clauses) then a typical k MCSAT instance will be *locked*, meaning that solutions are not only isolated, but separated pairwise by $O(\log n)$ bit flips [38]. By contrast, when k is odd there can be local rearrangements that connect solutions, and large clusters of closely related (in Hamming space) solutions are to be expected. Large sets of this type are undesirable in the construction of SAT filters, because two closely related solutions will only store slightly more information than one solution alone.

4.3 Bipartite 4FLSAT

To this point, each SAT variant presented operates implicitly over the complete graph K_n . That is, any set of k variables can be chosen to be in a clause. Further, there is yet another perspective on NAE3SAT: a clause can be thought of as a *frustrated loop* (cycle, in graph theoretic terms) in the Ising model. Each NAE3SAT clause is represented in the Ising model by a cycle of couplings with values ± 1 , containing an odd number of antiferromagnetic (positive, repulsive) couplings. So a random NAE3SAT clause is generated as a random frustrated 3-cycle from the complete graph.

In a new variant called *bipartite 4FLSAT* ($B4FLSAT$), each clause is a frustrated 4-cycle in the complete bipartite graph $K_{\frac{n}{2}, \frac{n}{2}}$. These instances are closely related to the frustrated loop instances that were the subject of recent benchmarking work [15, 19] where the instances are constructed in the native Chimera topology using frustrated loops of varying length. A $B4FLSAT$ clause is satisfied if and only if the Ising state corresponding to the truth assignment minimizes the energy of the clause in the Ising model.

$$\min_{s_1, s_2, s_3, s_4 \in \{-1, 1\}} (-s_1 s_2 - s_2 s_3 - s_3 s_4 + s_1 s_4).$$

Explicitly, the truth assignments satisfying a $B4FLSAT$ clause with no negated literals are TTTT, TTTF, TTFF, TFFF, FFFF, FFFT, FFFT, and FTTT. A 4FLSAT clause has a false positive rate of $1/2$. $B4FLSAT$ instances are unlocked, like $kMCSAT$ for odd k .

4.4 Threshold Analysis

Despite the small size of the problems investigated here, it is useful to understand the asymptotic structure of the solution space that governs the quality of attainable filters and scaling of sampling methods. A common feature of the SAT variants described here is the presence of a *satisfiability* transition and a *dynamical* transition. The satisfiability threshold is the ratio α_s of constraints to variables separating, asymptotically almost surely, regimes with and without solutions. The satisfiability threshold thus implies a bound \mathcal{E}_s on the maximum (asymptotically achievable) efficiency, since any filter built from an unsatisfiable instance will have false positive rate 1. The dynamical transition is, by contrast, the ratio α_d at which the solution space shatters into disconnected components. This shattering is related to hardness for sampling and optimization, although the relationship is complicated and an active area of research [22].

A simple rigorous upper bound on the satisfiability threshold (α_{1MM}) is obtained by a first moment method [27]; equivalently, α_{1MM} gives the information-theoretic upper bound on α_s in the context of filters. Transitions are also approximable by statistical physics techniques, namely the *1RSB energetic cavity method* and the *reconstruction on trees method* [26, 27]. The thresholds indicate that building efficient SAT filters requires sampling in the shattered phase, where local search methods will often struggle. Table 1 gives estimates on the transitions and maximum efficiency of NAE3SAT, 3in6SAT, and $B4FLSAT$.

Table 1. Thresholds and efficiency of the three CSPs studied in Section 6

Problem	Dynamical α_d	Satisfiability α_s	Annealed α_{1MM}	Efficiency \mathcal{E}_s
NAE3SAT	1.50	2.11	2.409	0.88
3in6SAT	0.48	0.57	0.596	0.96
B4FLSAT	0.50	0.78	1	0.78

4.5 Software Solvers

The next sections compare performance of the D-Wave processor against three classical software approaches. The first two, WalkSAT [33] and Dimetheus [12, 13], are SAT solvers, and are applied directly to the SAT problems in question (after naive clause-wise conversion to CNF). The third, Selby’s solver [32], is a specialized implementation of the Hamze-de Freitas-Selby algorithm, written to solve Ising problems on D-Wave’s native Chimera architecture. The version used, like in previous work [19], is modified to make it act more analogously to the D-Wave processor.

Both SAT solvers were run naively with the aim of generating 1000 solutions to each SAT instance. WalkSAT was run with default command line parameters and 1000 random restarts. Dimetheus was run 1000 times with command line parameters `-guide 0` and `-cdclSelectDirRule 0`. For both Selby and DW2, samples were mapped from the embedded Chimera space to the SAT space via majority vote, then quenched to a local minimum (see [20] for further explanation).

5 Case Study 1: A Blocked MNAE3SAT Filter

This section describes the construction of a 3500-block MNAE3SAT for 262,144 randomly generated 16-byte numbers (UUIDs). The MNAE3SAT instance for each block was constructed over 45 variables, giving a mean clause-to-variable ratio of $\bar{\alpha} \approx 1.664$. This is very close to the minimum requirement for matching the efficiency of a Bloom filter (ignoring small-system MNAE3SAT considerations discussed in Section 4), i.e. $\log(2)/\log_2(3/4)$. Of the 3500 MNAE3SAT instances, 77 were unsatisfiable.

For each MNAE3SAT instance, 24,000 (not necessary optimal) samples were drawn from DW2, 10,000 from Selby, and 1000 each from WalkSAT and Dimetheus. WalkSAT and Dimetheus successfully returned 1000 solutions for each satisfiable instance, while the number of solutions returned by DW2 and Selby varied substantially (see Fig. 2). Variation in the numbers reflect different operational modes of the algorithms, and the probability of each solver’s output of being a valid solution (low for DW2 and Selby). In order to avoid conditioning filter construction on the number of solutions returned, each instance drew from a set of solutions whose size was limited by the minimum number of solutions returned by any solver. For most blocks this was Dimetheus and WalkSAT.

Fig. 3 shows the distribution of α and the number of solutions for these blocks (computed using sharpSAT [34]), highlighting a challenge in constructing

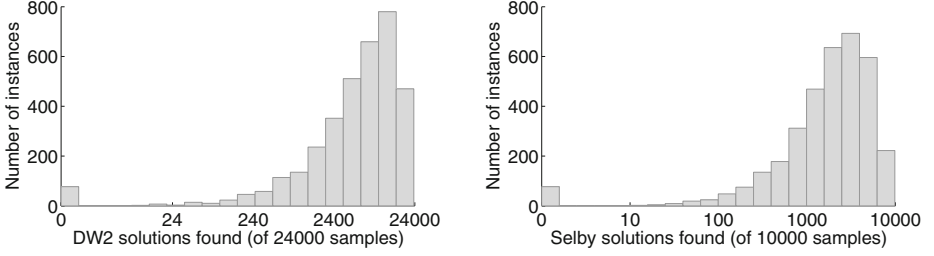


Fig. 2. Solutions returned by DW2 (left) and Selby (right)

efficient blocked filters. In order to ensure that almost all blocks are satisfiable, the number of blocks must be high enough that almost the entire binomial distribution of α is below the satisfiability threshold α_s . The effect seen here declines in importance as the size of blocks increases, and can be mitigated in several ways that trade efficiency for query time or similar [21]. This paper does not explore the question further.

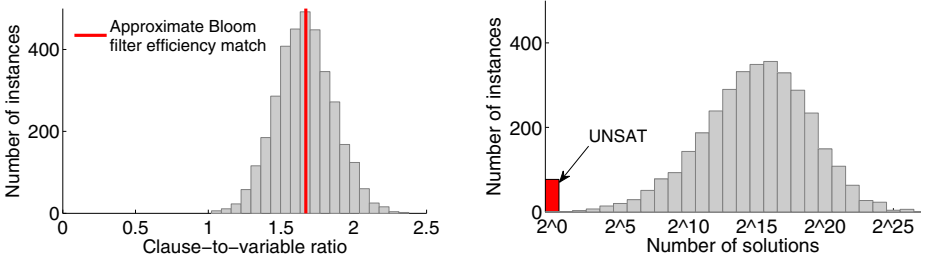


Fig. 3. Clause-to-variable ratios (left) and solution counts (right) for 3500 MNAE3SAT instances. A line indicates $-\log(2)/\log_2(3/4)$, the ratio at which a single-solution NAE3SAT filter matches the optimal efficiency of a Bloom filter

Fig. 4 shows the decline in efficiency for the overall filter as more and more solutions are stored per block, up to 20. Two methods of selecting r SAT solutions for a filter block are used. In the *online* approach, r random solutions are selected from the multiset of solutions returned by the solver in question. This reflects the situation where the solver returns solutions that are independent of the previously returned solutions, and the user wants r solutions as fast as possible. In the *greedy offline* approach, each solution is iteratively selected subject to minimizing the false positive rate of the filter at that point. This reflects the more realistic situation in which the user spends a certain amount of time generating SAT solutions, then constructs a filter greedily with a subset of the solutions at hand.

It is interesting that in this context Dimetheus starts out strong and declines in efficacy, while the opposite is true for WalkSAT. For online filter construction,

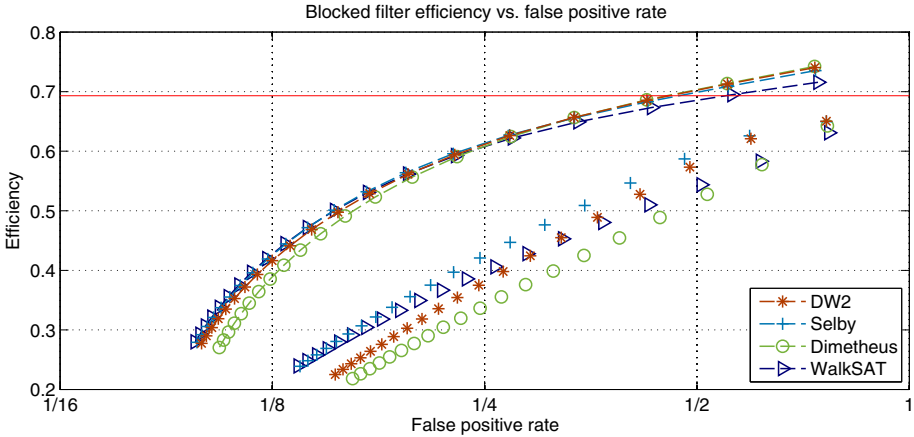


Fig. 4. (Color online) Blocked filter efficiency versus false positive rate for MNAE3SAT filter at mean clause-to-variable ratio of 1.66. Upper data with lines represents offline (greedy) filter construction. Lower data represents online filter construction. The efficiency of an ideal Bloom filter, 0.69, is indicated

Selby offers a significant advantage over DW2. This can be explained by current calibration nonidealities in the processor, which tilt the processor in a certain direction in the Hamming space, reducing its ability to sample solutions equitably. All solvers can easily solve the instances in question; the next section includes a look at performance scaling for various CSPs.

6 Case Study 2: Filter and Performance Scaling

The second part of the case study investigates how the construction of filters evolves as SAT instances grow. Considered here are random NAE3SAT, 3in6SAT, and B4FLSAT instances of increasing size. For each CSP the testbed contains 10 instances of each size shown, using only even sizes for B4FLSAT. Each instance is generated near the midpoint between the dynamical and satisfiability thresholds; NAE3SAT, 3in6SAT, and B4FLSAT use target ratios of 1.8, 0.55, and 0.70, respectively. For each size of each SAT variant, the ten instances were used as blocks in the construction of a blocked SAT filter.

For each SAT instance in this section, only 20,000 samples were drawn using DW2, representing a total anneal time of 0.4s. As before, 10,000 samples were drawn using Selby, and 1000 were drawn using each of WalkSAT and Dimetheus. False positive rates for the filters were estimated using a Monte Carlo approach.

Fig. 5 shows the evolution of filter efficiency versus false positive rate as the number of solutions used for each block increases from 1 to 20. The dependence on the properties of the CSP used are clear. 3in6SAT filters should show potential for very efficient filters that suffer from a low clause-to-variable ratio and the requirement of solving a hard CSP, but large filters reflect the low number of

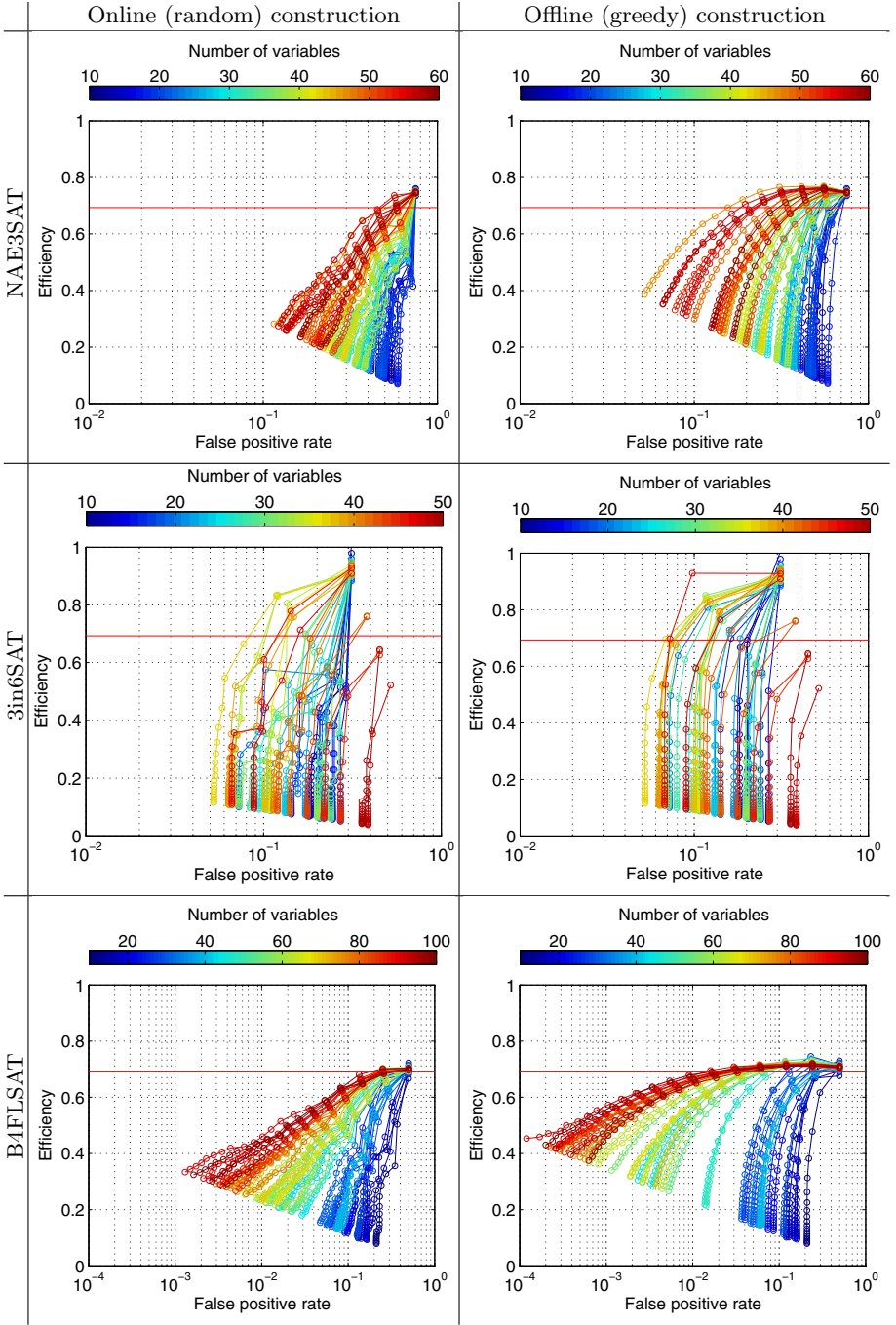


Fig. 5. (Color online) Evolution of filter efficiency/false positive rate tradeoff for filters constructed by DW2.

solutions returned by DW2 and more so Selby on these instances. Although the B4FLSAT filters maintain a relatively consistent false positive rate, these filters are bound to have efficiency at most 0.78, as discussed in Section 4.4.

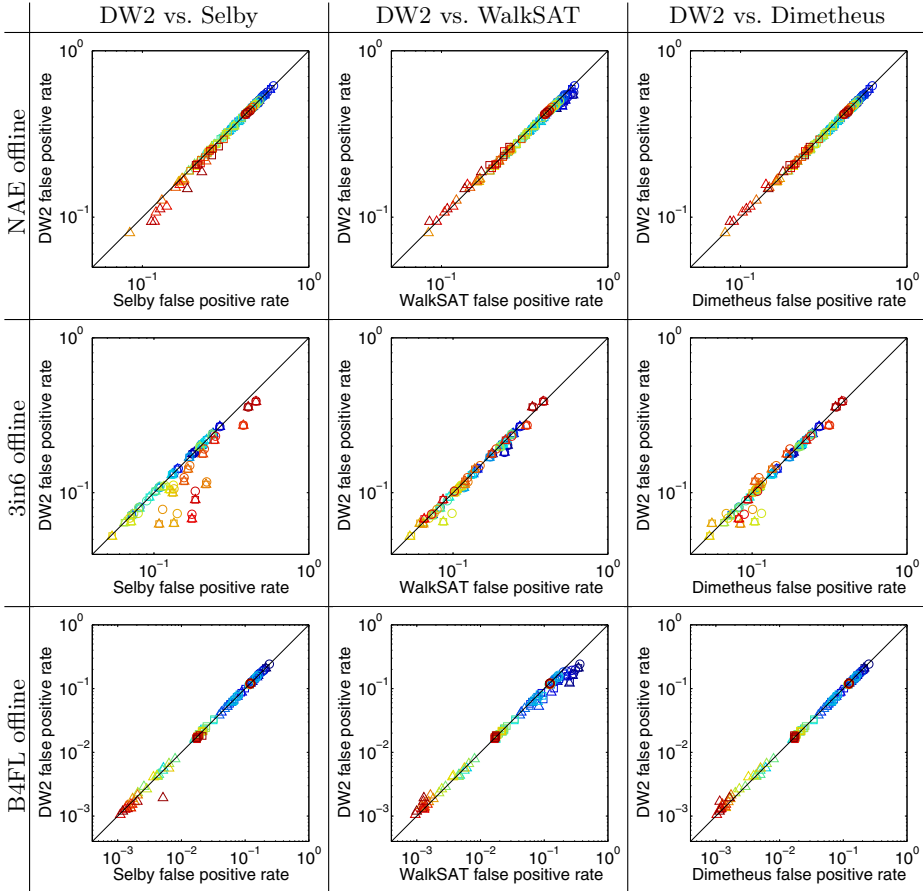


Fig. 6. (Color online) Instance-wise comparison of false positive rates for DW2 versus classical software solvers for offline filter construction. Filters using 3, 6, and 12 solutions are denoted by \circ , \square , and \triangle respectively. Colors/shades denote problem size as in Fig. 5. Points below the diagonal favor DW2.

Fig. 6 gives a direct comparison of false positive rate for offline filters generated using output from DW2 versus each classical solver, using 3, 6, and 12 solutions per block. WalkSAT’s relatively poor performance on small B4FLSAT instances seems to be a bona fide weakness of the solver as a sampler on these small systems, which may have troublesome structural characteristics. False positive rates for DW2 would likely improve given refined calibration, which is currently underway. Overall, the four solvers give results of comparable quality.

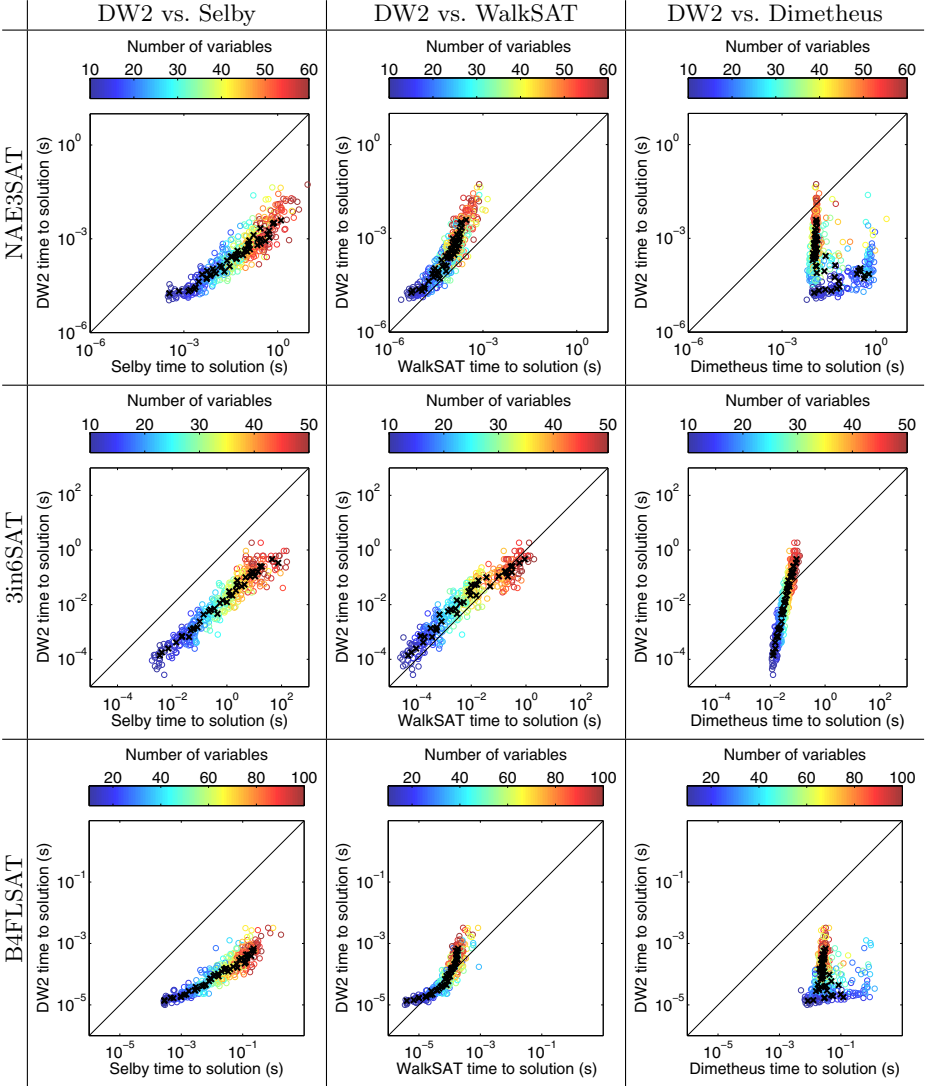


Fig. 7. (Color online) Instance-wise comparison of solution time for DW2 versus classical software solvers. \times markers indicate median for a given size.

6.1 Scaling of Solution Time

Fig. 7 shows how the time required by each solver to draw a solution scales with the number of variables. Dimetheus, being a multi-mode solver that sometimes resorts to exhaustive search, is likely entering this slow mode prematurely for small instances of NAE3SAT and B4FLSAT. Solution time of DW2 versus WalkSAT is particularly interesting for 3in6SAT, as explained in the conclusion. Although both solvers find the B4FLSAT instances easy, it would be interesting to know if there

is an explanation behind the sharp inflection point in the lower-middle panel. The analysis here is quite superficial, but serves to give a counterpoint to Fig. 5 from the perspective of problem difficulty rather than filter efficiency.

7 Conclusions

The case study of 3500 MNAE3SAT instances provides evidence that DW2 is capable of constructing filters competitively, but solution of larger instances is required to improve efficiency. NAE3SAT, at these block sizes, can only give high efficiency at high false positive rates. Furthermore, NAE3SAT filter efficiency reaches an information-theoretic limit at 0.88. This motivates the study of more exotic CSPs in the Ising model. k MCSAT filters for $k \geq 4$ and B4FLSAT filters both show advantages and weaknesses when compared with NAE3SAT. Tradeoffs are between difficulty, maximum efficiency at the satisfiability phase transition, maintenance of efficiency as false positive rate drops, and block size relative to keyword count.

The fact that DW2 appears to show a scaling advantage over WalkSAT (see Fig. 7, center panel) for this limited set of small 3in6SAT instances raises the question of whether or not multi-qubit cotunneling could play a role for large-clause MAX-CUT CSPs. Embedded 3in6SAT problems may be a good place to look for instances with long relaxation times, i.e. where an advantage might be gained by lengthening the anneal time from 20 μ s. To this point, finding such instances has been a challenge [15, 31, 35].

The ability of a SAT filter to drill down to a low false positive rate is a reflection of the global richness of the solution space. This study shows the limitations of small-blocked filters, and points to k MCSAT filters for larger k as a possible avenue of research.

Acknowledgments. The authors are very grateful to Sean Weaver for helpful discussions and advice regarding this work and to the anonymous referees for their careful readings.

Open Access This chapter is distributed under the terms of the Creative Commons Attribution Noncommercial License, which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Albash, T., Vinci, W., Mishra, A., Warburton, P.A., Lidar, D.A.: Consistency tests of classical and quantum models for a quantum annealer. *Physical Review A* **91**(4), 042314 (2015)
2. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM* **13**(7), 422–426 (1970)

3. Boixo, S., Smelyanskiy, V.N., Shabani, A., Isakov, S.V., Dykman, M., Denchev, V.S., Amin, M., Smirnov, A., Mohseni, M., Neven, H.: Computational role of collective tunneling in a quantum annealer. arXiv preprint [arXiv:1411.4036](https://arxiv.org/abs/1411.4036) (2014)
4. Bunyk, P., Hoskinson, E., Johnson, M., Tolkacheva, E., Altomare, F., Berkley, A., Harris, R., Hilton, J., Lanting, T., Przybysz, A., et al.: Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity* (2014)
5. Cai, J., Macready, W., Roy, A.: A practical heuristic for finding graph minors. arXiv preprint [arXiv:1406.2741](https://arxiv.org/abs/1406.2741) (2014)
6. Chakraborty, S., Meel, K.S., Vardi, M.Y.: A scalable and nearly uniform generator of SAT witnesses. In: Sharygina, N., Veith, H. (eds.) *CAV 2013*. LNCS, vol. 8044, pp. 608–623. Springer, Heidelberg (2013)
7. Choi, V.: Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Information Processing* **7**(5), 193–209 (2008)
8. Crawford, J.M., Auton, L.D.: Experimental results on the crossover point in random 3-SAT. *Artificial Intelligence* **81**(1), 31–57 (1996)
9. Dickson, N., et al.: Thermally assisted quantum annealing of a 16-qubit problem. *Nature Communications* **4**, May 1903, January 2013. <http://www.ncbi.nlm.nih.gov/pubmed/23695697>
10. Diestel, R.: *Graph Theory*. Graduate Texts in Mathematics, vol. 173, 4th edn. Springer (2012)
11. Fan, B., Andersen, D.G., Kaminsky, M., Mitzenmacher, M.D.: Cuckoo filter: practically better than Bloom. In: *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, pp. 75–88. ACM (2014)
12. Gableske, O.: Dimetheus. In: *SAT Competition 2014: Solver and Benchmark Descriptions*, pp. 29–30 (2014)
13. Gableske, O.: An Ising model inspired extension of the product-based MP framework for SAT. In: Sinz, C., Egly, U. (eds.) *SAT 2014*. LNCS, vol. 8561, pp. 367–383. Springer, Heidelberg (2014)
14. Garey, M.R., Johnson, D.S.: *Computers and Intractability*. W.H. Freeman (1979)
15. Hen, I., Albash, T., Job, J., Rønnow, T.F., Troyer, M., Lidar, D.: Probing for quantum speedup in spin glass problems with planted solutions (2015). arXiv preprint [arXiv:1502.01663v2](https://arxiv.org/abs/1502.01663v2)
16. Jerrum, M., Valiant, L., Vazirani, V.: Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43**, 169–188 (1986)
17. Jiménez, A., Kiwi, M.: Computational hardness of enumerating groundstates of the antiferromagnetic Ising model in triangulations. *Discrete Applied Mathematics* (2014)
18. Johnson, M., Amin, M., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A., Johansson, J., Bunyk, P., et al.: Quantum annealing with manufactured spins. *Nature* **473**(7346), 194–198 (2011)
19. King, A.D.: Performance of a quantum annealer on range-limited constraint satisfaction problems (2015). arXiv preprint [arXiv:1502.02098v1](https://arxiv.org/abs/1502.02098v1)
20. King, A.D., McGeoch, C.C.: Algorithm engineering for a quantum annealing platform. arXiv preprint [arXiv:1410.2628](https://arxiv.org/abs/1410.2628) (2014)
21. Krimer, E., Erez, M.: The power of $1+\alpha$ for memory-efficient Bloom filters. *Internet Mathematics* **7**(1), 28–44 (2011)

22. Krzakala, F., Zdeborová, L.: Phase transitions and computational difficulty in random constraint satisfaction problems. In: *Journal of Physics: Conference Series*, vol. 95, p. 012012. IOP Publishing (2008)
23. Lovász, L.: Coverings and colorings of hypergraphs. In: *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pp. 3–12. Utilitas Mathematica Publishing, Winnipeg (1973)
24. Lovett, S., Porat, E.: A lower bound for dynamic approximate membership data structures. In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 797–804. IEEE (2010)
25. Lucas, A.: Ising formulations of many NP problems. *Frontiers in Physics* **2**(5) (2014)
26. Mézard, M., Montanari, A.: Reconstruction on trees and spin glass transition. *Journal of Statistical Physics* **124**(6), 1317–1350 (2006)
27. Mézard, M., Montanari, A.: *Information, Physics, and Computation*. Oxford University Press (2009)
28. Pagh, A., Pagh, R., Rao, S.S.: An optimal Bloom filter replacement. In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 823–829. Society for Industrial and Applied Mathematics (2005)
29. Porat, E.: An optimal Bloom filter replacement based on matrix solving. In: Frid, A., Morozov, A., Rybalchenko, A., Wagner, K.W. (eds.) *CSR 2009. LNCS*, vol. 5675, pp. 263–273. Springer, Heidelberg (2009)
30. Putze, F., Sanders, P., Singler, J.: Cache-, hash- and space-efficient Bloom filters. In: Demetrescu, C. (ed.) *WEA 2007. LNCS*, vol. 4525, pp. 108–121. Springer, Heidelberg (2007)
31. Rønnow, T., Wang, Z., Job, J., Boixo, S., Isakov, S., Wecker, D., Martinis, J., Lidar, D., Troyer, M.: Defining and detecting quantum speedup. *Science* **345**(6195), 420–424 (2014)
32. Selby, A.: Efficient subgraph-based sampling of Ising-type models with frustration. arXiv preprint [arXiv:1409.3934v1](https://arxiv.org/abs/1409.3934v1) (2014)
33. Selman, B., Kautz, H., Cohen, B., et al.: Local search strategies for satisfiability testing. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge* **26**, 521–532 (1993)
34. Thurley, M.: sharpSAT – counting models with advanced component caching and implicit BCP. In: Biere, A., Gomes, C.P. (eds.) *SAT 2006. LNCS*, vol. 4121, pp. 424–429. Springer, Heidelberg (2006)
35. Venturelli, D., Mandrà, S., Knysh, S., O’Gorman, B., Biswas, R., Smelyanskiy, V.: Quantum optimization of fully-connected spin glasses. arXiv preprint [arXiv:1406.7553](https://arxiv.org/abs/1406.7553) (2014)
36. Walker, A.: Filters. Undergraduate thesis, Haverford College, Haverford, PA (2007)
37. Weaver, S.A., Ray, K.J., Marek, V.W., Mayer, A.J., Walker, A.K.: Satisfiability-based set membership filters. *Journal on Satisfiability, Boolean Modeling and Computation* **8**, 129–148 (2014)
38. Zdeborová, L., Mézard, M.: Locked constraint satisfaction problems. *Physical Review Letters* **101**(7), 078702 (2008)